

What is Claimed:

1. A method for enhancing a Precomputed Radiance Transfer (PRT) mesh comprising a tessellation of triangles, said triangles comprising vertices and edges, said method comprising dividing at least one triangle in the mesh, but not all triangles in the mesh, into at least two or more triangles apiece.
2. The method of claim 1, wherein said dividing at least one triangle comprises dividing at least one edge and creating at least one new vertex and at least one new edge running from said vertex for said triangle.
3. The method of claim 2 further comprising the following:
 - sampling a PRT vector at each vertex and each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and
 - for each edge having an error at its midpoint that is greater than a predetermined error tolerance value (a “marked edge”), dividing the triangle for that edge into at least two new triangles as follows:
 - dividing the marked edge into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and
 - creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”).
4. The method of claim 3 further comprising, for each new edge, including divided edges and common edges, recursively performing the following steps for each such new edge created (herein below, each an “edge”):
 - sampling a PRT vector each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each edge having an error at its midpoint that is greater than the predetermined error tolerance value (a “marked edge”), dividing the triangle for that edge into at least two new triangles as follows:

dividing the marked edge into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and

creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”);
said step recursively performed until either (a) no new edges are created in preceding recursive iteration or (b) a predetermined number of recursions have been performed.

5. The method of claim 4 wherein a triangle, consisting of one edge having an error at its midpoint that is greater than the predetermined error tolerance value, is divided into at least two new triangles.

6. The method of claim 4 wherein a triangle, consisting of two edges having errors at the midpoints thereof greater than the predetermined error tolerance value, is divided into three new triangles.

7. The method of claim 4 wherein a triangle, consisting of three edges having errors at the midpoints thereof greater than the predetermined error tolerance value, is divided into four new triangles.

8. The method of claim 2 further comprising:

sampling a PRT vector at each vertex and each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value (a “marked edge”), dividing said triangle into four new triangles as follows:

dividing each edge of said triangle into two new edges with a new vertex at the midpoint for said such edge (each a “divided edge”); and
creating new edges between each pair of new vertices (each a “common edge”).

9. The method of claim 8 further comprising, for each new edge, including divided edges and common edges, recursively performing the following steps for each such new edge created (herein below, each an “edge”):

sampling a PRT vector at each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value, dividing said triangle into four new triangles as follows:

dividing each edge into two new edges with a new vertex at the midpoint for said such edge; and

creating new edges between each pairs of new vertices;
said steps recursively performed until either (a) no new edges are created in preceding recursive iteration or (b) a predetermined number of recursions have been performed.

10. A system for enhancing a Precomputed Radiance Transfer (PRT) mesh comprising a tessellation of triangles, said triangles comprising vertices and edges, said system comprising a subsystem for dividing at least one triangle in the mesh, but not all triangles in the mesh, into at least two or more triangles apiece.

11. The system of claim 10 wherein said subsystem for dividing at least one triangle in the mesh comprises a subsystem for dividing at least one edge and creating at least one new vertex and at least one new edge running from said vertex.

12. The system of claim 11 further comprising:

a subsystem for sampling a PRT vector at each vertex and each midpoint of each edge;

a subsystem for computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

a subsystem for dividing a marked edge, that is, an edge having an error at its midpoint that is greater than a predetermined error tolerance value, into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and

a subsystem for creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”).

13. The system of claim 12 further comprising, for each new edge, including divided edges and common edges, a subsystem for recursively performing the following steps for each such new edge created (herein below, each an “edge”):

sampling a PRT vector each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each edge having an error at its midpoint that is greater than the predetermined error tolerance value (a “marked edge”), dividing the triangle for that edge into at least two new triangles as follows:

dividing the marked edge into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and

creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”);
said step recursively performed until either (a) no new edges are created in preceding recursive iteration or (b) a predetermined number of recursions have been performed.

14. The system of claim 13 wherein a triangle, consisting of one edge having an error at its midpoint that is greater than the predetermined error tolerance value, is divided into at least two new triangles.

15. The system of claim 13 wherein a triangle, consisting of two edges having errors at the midpoints thereof greater than the predetermined error tolerance value, is divided into three new triangles.

16. The system of claim 13 wherein a triangle, consisting of three edges having errors at the midpoints thereof greater than the predetermined error tolerance value, is divided into four new triangles.

17. The system of claim 11 further comprising:

a subsystem for sampling a PRT vector at each vertex and each midpoint of each edge;
a subsystem for computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and
a subsystem for dividing each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value into four new triangles as follows:

dividing each edge into two new edges with a new vertex at the midpoint for said such edge (each a “divided edge”); and

creating new edges between each pairs of new vertices (each a “common edge”).

18. The system of claim 17 further comprising a new subsystem that, for each new edge, including divided edges and common edges, recursively performs the following steps for each such new edge created (herein below, each an “edge”):

sampling a PRT vector at each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value, dividing said triangle into four new triangles as follows:

dividing each edge into two new edges with a new vertex at the midpoint for said such edge; and

creating new edges between each pairs of new vertices;

said steps recursively performed until either (a) no new edges are created in preceding recursive iteration or (b) a predetermined number of recursions have been performed.

19. A computer-readable medium comprising computer-readable instructions for enhancing a Precomputed Radiance Transfer (PRT) mesh comprising a tessellation of triangles, said triangles comprising vertices and edges, said computer-readable instructions comprising instructions for dividing at least one triangle in the mesh, but not all triangles in the mesh, into at least two or more triangles apiece.
20. The computer-readable instructions of claim 19 wherein said instructions for dividing at least one triangle in the mesh comprise instructions for dividing at least one edge and creating at least one new vertex and at least one new edge running from said vertex for a triangle.
21. The computer-readable instructions of claim 20 further comprising instructions for:
sampling a PRT vector at each vertex and each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and
for each edge having an error at its midpoint that is greater than a predetermined error tolerance value (a “marked edge”), dividing the triangle for that edge into at least two new triangles as follows:
dividing the marked edge into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and
creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”).
22. The computer-readable instructions of claim 21 further comprising instructions for recursively performing the following steps for each new edge, including divided edges and common edges (herein below, each an “edge”):

sampling a PRT vector each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each edge having an error at its midpoint that is greater than the predetermined error tolerance value (a “marked edge”), dividing the triangle for that edge into at least two new triangles as follows:

dividing the marked edge into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and

creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”);
said step recursively performed until either (a) no new edges are created in preceding recursive iteration or (b) a predetermined number of recursions have been performed.

23. The computer-readable instructions of claim 22 further comprising instructions for dividing a triangle into at least two new triangles when said triangle consists of one edge having an error at its midpoint that is greater than the predetermined error tolerance value.

24. The computer-readable instructions of claim 22 further comprising instructions for dividing a triangle into three new triangles when said triangle consists of two edges having errors at midpoints that are greater than the predetermined error tolerance value.

25. The computer-readable instructions of claim 22 further comprising instructions for dividing a triangle into four new triangles when said triangle consists of three edges having errors at midpoints that are greater than the predetermined error tolerance value.

26. The computer-readable instructions of claim 20 further comprising instructions for:
sampling a PRT vector at each vertex and each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value (a “marked edge”), dividing said triangle into four new triangles as follows:

dividing each edge of said triangle into two new edges with a new vertex at the midpoint for said such edge (each a “divided edge”); and
creating new edges between each pair of new vertices (each a “common edge”).

27. The computer-readable instructions of claim 26 further comprising instructions for recursively performing the following steps for each new edge created, including divided edges and common edges (herein below, each an “edge”):

sampling a PRT vector at each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and

for each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value, dividing said triangle into four new triangles as follows:

dividing each edge into two new edges with a new vertex at the midpoint for said such edge; and

creating new edges between each pairs of new vertices;
said steps recursively performed until either (a) no new edges are created in preceding recursive iteration or (b) a predetermined number of recursions have been performed.

28. A hardware control device comprising means for enhancing a Precomputed Radiance Transfer (PRT) mesh comprising a tessellation of triangles, said triangles comprising vertices and edges, said computer-readable instructions comprising instructions for dividing at least one triangle in the mesh, but not all triangles in the mesh, into at least two or more triangles apiece by dividing at least one edge and creating at least one new vertex and at least one new edge running from said vertex.

29. The hardware control device of claim 28 further comprising means for:

- sampling a PRT vector at each vertex and each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and
- for each edge having an error at its midpoint that is greater than a predetermined error tolerance value (a “marked edge”), dividing the triangle for that edge into at least two new triangles as follows:
 - dividing the marked edge into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and
 - creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”).

30. The hardware control device of claim 29 further comprising means for recursively performing the following steps for each new edge, including divided edges and common edges (herein below, each an “edge”):

- sampling a PRT vector each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and
- for each edge having an error at its midpoint that is greater than the predetermined error tolerance value (a “marked edge”), dividing the triangle for that edge into at least two new triangles as follows:
 - dividing the marked edge into two new edges with a new vertex at the midpoint for said edge (each a “divided edge”); and
 - creating a new edge between said new vertex and a vertex not on said edge, such that said edge is common to said two new triangles (each a “common edge”);
- said step recursively performed until either (a) no new edges are created in preceding recursive iteration or (b) a predetermined number of recursions have been performed.

31. The hardware control device of claim 28 further comprising means for:
 - sampling a PRT vector at each vertex and each midpoint of each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and
 - for each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value (a “marked edge”), dividing said triangle into four new triangles as follows:
 - dividing each edge of said triangle into two new edges with a new vertex at the midpoint for said such edge (each a “divided edge”); and
 - creating new edges between each pair of new vertices (each a “common edge”).
32. The hardware control device of claim 31 further comprising means for recursively performing the following steps for each new edge created, including divided edges and common edges (herein below, each an “edge”):
 - sampling a PRT vector at each edge, and computing an error at each midpoint at each edge based on the PRT vector for each said midpoint and the PRT vectors for each vertex corresponding to each said edge; and
 - for each triangle comprising at least one edge having an error at its midpoint that is greater than a predetermined error tolerance value, dividing said triangle into four new triangles as follows:
 - dividing each edge into two new edges with a new vertex at the midpoint for said such edge; and
 - creating new edges between each pairs of new vertices;

[Remainder of Page Intentionally Left Blank]